嵌入式 **linux** 工程师面试题目 **C** 语言基础部分

5trnga 发布于 2008-5-23 | 1059 次阅读 字号: 大 中 小 (网友评论 1 条) 我要评论

在网上看到了一篇嵌入式的面试题，忽然觉得自己很渺小……在未来一段时间内我会陆续贴上我的答案，当然可能有些我做不出来，做出来的也不一定准确，在检验和提高自己的同时，希望有走过路过高人指点一二。

下面的题目必须全部答对才给分：

1、 如何在 C 中初始化一个字符数组。

2、 如何在 C 中为一个数组分配空间。

3、 如何初始化一个指针数组。

4、如何定义一个有 10 个元素的整数型指针数组。

5、 s[10]的另外一种表达方式是什么。

6、 GCC3.2.2 版本中支持哪几种编程语言。

7、 要使用 CHAR_BIT 需要包含哪个头文件。

8、 对(-1.2345)取整是多少？

9、 如何让局部变量具有全局生命期。

10、C 中的常量字符串应在何时定义？

11、如何在两个.c 文件中引用对方的变量。

12、使用 malloc 之前需要做什么准备工作。

13、realloc 函数在使用上要注意什么问题。

14、strtok 函数在使用上要注意什么问题。

15、gets 函数在使用上要注意什么问题。

16、C 语言的词法分析在长度规则方面采用的是什么策略？

17、a+++++b 所表示的是什么意思？有什么问题？

18、如何定义 Bool 变量的 TRUE 和 FALSE 的值。

19、C 语言的 const 的含义是什么。在定义常量时，为什么推荐使用 const，而不是#define。

20、C 语言的 volatile 的含义是什么。使用时会对编译器有什么暗示。

这部分是 ANSI C 的一些问题，题目的前提是必须都答对，看似很变态，但是细想一下，这些都是最基础的，虽然我们在使用他们的时候会犯这样那样的错误，但是最终目的是不犯错误，不是么，那么好，从最基础的开始。

1、 如何在 C 中初始化一个字符数组。

这个问题看似很简单，但是我们要将最简单的问题用最严谨的态度来对待。关键的地方：初始化、字符型、数组。最简单的方法是 char array[];。这个问题看似解决了，但是在初始化上好像还欠缺点什么，个人认为：char array[5]={'1','2','3','4','5'};或者 char array[5]={"12345"};或者 char array[2][10]={"China","Beijing"};也许更符合"初始化"的意思。

2、 如何在 C 中为一个数组分配空间。

docin 豆丁 www.docin.com

最简单的方法是：char array[5]；意思是分配给数组 array 一个 5 个字节的空间。但是我们要知道在 C 中数组其实就是一个名字，其实质含义就是指针，比如 char array[]；是到底分配的多少空间？所以我们要将其分成为两种不同的形式给出答案：

一种是栈的形式：char array[5]；

一种是堆的形式：char *array; array=(char *)malloc(5);//C++: array=new char[5];

堆和栈的含义其实我也没弄太透彻，改天明白了再发一篇。

我们要明白的是，第一种形式空间分配的大小可能会受操作系统的限制,比如 windows 会限制在 2M；第二种形式成空间分配很灵活，想分配多少分配多少，只要 RAM 够大。

3、 如何初始化一个指针数组。

首先明确一个概念，就是指向数组的指针，和存放指针的数组。

指向数组的指针：char (*array)[5];含义是一个指向存放 5 个字符的数组的指针。

存放指针的数组：char *array[5];含义是一个数组中存放了 5 个指向字符型数据的指针。

按照题意，我理解为初始化一个存放指针的数组，char *array[2]={"China","Beijing"}；其含义是初始化了一个有两个指向字符型数据的指针的数组，这两个指针分别指向字符串"China"和"Beijing"。

4、如何定义一个有 10 个元素的整数型指针数组。

既然只是定义而不是初始化，那就很简单且没有争议了：int *array[10];。

5、 s[10]的另外一种表达方式是什么。

前面说过了，数组和指针其实是数据存在形态的两种表现形式，如果说对于数组 s[]，我们知道*s=s[0]，那么 s[10]的另一种表达方式就是：*(s+10)。

6、 GCC3.2.2 版本中支持哪几种编程语言。

这个问题实在变态，就像问你#error 的作用是什么一样。不可否认，gcc 是 linux 下一个亮点，是一个备受无数程序员推崇的编译器，其优点省略 1000 字，有兴趣可以自己查，我翻了翻书，书上曰：支持 C,C++,Java,Obj-C,Ada,Fortran,Pascal,Modula-3 等语言，这个"等"比较要命，不过我认为已经很全了，如果认为还是不全，干脆把 ASM 也加上算了，不过那已经不算是编译了。

7、 要使用 CHAR_BIT 需要包含哪个头文件。

如果结合上面的问题，答题的人估计会认为自己撞鬼了，这个问题实在是……搜索了一下，应该是 limits.h。

8、 对(-1.2345)取整是多少？

docin 豆丁 www.docin.com

其实不同的取整函数可能有不同的结果，不过这个数没有太大的争议，答案是-1。

9、 如何让局部变量具有全局生命期。

具体的生命期的概念我觉得我还要好好深入的学习一下，但是这个题目还算比较简单，即用 static 修饰就可以了，但是只是生命期延长，范围并没有扩大，除非把这个变量定义在函数体外的静态区，不过那样就变成全局变量了，仿佛不符合题目要求。

10、C 中的常量字符串应在何时定义？

这个问题说实话不是很理解题干的意思，据我理解，有两种情况，一种是预处理阶段，用#define 定义；还有就是使用 const 修饰词，不过 const 修饰的是一个变量，其含义是"只读"，称之为常量并不准确，但是确实可以用操作变量的方法当常量用。所以还是第一种比较靠谱。

11、如何在两个.c 文件中引用对方的变量。

这个问题也问的挺含糊的，怎么说呢，最简单最直接的方法是为变量添加 extern 修饰词，当然，这个变量必须是全局变量。还有一种就是利用函数调用来进行变量的间接引用，比如这个 C 文件中的一个函数引用另外一个 C 中的函数，将变量通过实参的形式传递过去。不过题目既然说是引用，那么还是用第一个答案好了。

12、使用 malloc 之前需要做什么准备工作。

其实准备工作很多啊，比如你需要一台计算机之类的。玩笑话，我们首先要知道 malloc 的用途，简单的说就是动态的分配一段空间，返回这段空间的头指针。实际的准备工作可以这么分：需要这段空间的指针是否存在，若不存在，则定义一个指针用来被赋值，还要清楚要返回一个什么类型的指针，分配的空间是否合理；如果指针已经存在，那么在重新将新的空间头地址赋值给这个指针之前，要先判断指针是否为 NULL，如果不是要 free 一下，否则原来的空间就会被浪费，或者出错，free 之后就按照前一种情形考虑就可以了。

13、realloc 函数在使用上要注意什么问题。

这个函数我也才知道的，汗一个。据我的初步理解，这个函数的作用是重新分配空间大小，返回的头指针不变，只是改变空间大小。既然是改变，就有变大、变小和为什么改变的问题。变大，要注意不能大到内存溢出；变小，那变小的那部分空间会被征用，原有数据不再存在；为什么改变，如果是想重新挪作他用，还是先 free 了吧。

14、strtok 函数在使用上要注意什么问题。

这个问题我不知道能不能回答全面，因为实在是用的很少。这个函数的作用是分割字符串，但是要分割的字符串不能是常量，这是要注意的。比如先定义一个字符串：char array[]="part1,part2";，strtok 的原形是 char *strtok(char *string, char *delim);，我们将","作为分隔符，先用 pt=strtok(array,",");，得到的结果 print 出来就是"part1"，那后面的呢，要写成 pt=strtok(NULL,",");，注意，要用 NULL，

如果被分割的字符串会被分成 N 段，那从第二次开始就一直要用 NULL。总结起来，需要注意的是：被分割的字符串和分隔符都要使用变量；除第一次使用指向字符串的指针外，之后的都要使用 NULL；注意使用这个函数的时候千万别把指针跟丢了，不然就全乱了。

15、gets 函数在使用上要注意什么问题。

这是一个键盘输入函数，将输入字符串的头地址返回。说到要注意的问题，我还是先查了一下网上的一些情况，需要注意的就是 gets 以输入回车结束，这个地球人都知道，但是很多人不知道的是，当你输入完一个字符串后，这个字符串可能依然存在于这个标准输入流之中，当再次使用 gets 的时候，也许会把上次输入的东西读出来，所以应该在使用之后用 fflush(stdin);处理一下，将输入流清空。最后也还是要注意溢出的问题。关于这个答案我比较含糊，不知道有没有高人高见？

16、C 语言的词法分析在长度规则方面采用的是什么策略？

我无语……闻所未闻啊……还是搜索了一下，有一篇文章，地址是：http://202.117.80.9/jp2005/20/kcwz/wlkc/wlkc/03/3_5_2.htm，是关于词法分析器的。其中提到了两点策略：(1) 按最长匹配原则确定被选的词型；(2) 如果一个字符串能为若干个词型匹配，则排列在最前面的词型被选中。不知道是不是题干的要求，还是其他什么。我乃一介草民，望达人指点迷津！

17、a+++++b 所表示的是什么意思？有什么问题？

这个东西（称之为东西一点都不过分）其实并没有语法错误，按照 C 对运算符等级的划分，++的优先级大于+，那么这句话会被编译器看做：(a++)+(++b)，这回明白了吧。有什么问题，语法上没有问题，有的是道德上的问题！作为一个优秀的程序员，我们要力求语句的合法性和可读性，如果写这句的人是在一个 team 里，那么他基本会被打的半死……最后讨论一下结果：假设 a 之前的值是 3，b 是 4，那么运行完这个变态语句后，a 的值是 4，b 是 5，语句的结果是 8。

www.docin.com

18、如何定义 Bool 变量的 TRUE 和 FALSE 的值。

不知道这个题有什么陷阱，写到现在神经已经大了，一般来说先要把 TURE 和 FALSE 给定义了，使用#define 就可以：

#define TURE 1

#define FALSE 0

如果有一个变量需要定义成 bool 型的，举个例子：bool a=TURE;就可以了。

19、C 语言的 const 的含义是什么。在定义常量时，为什么推荐使用 const，而不是#define。

首先，这个题干抽了 10 题回答的一个大嘴巴。关于常量的概念看来我要好好看看书了……我说过了，const 修饰词可以将一个变量修饰为"只读"，这个就能称为常量么？姑且认为可以。回到题目中，const 是只读的意思，它限定一个变量不允许被改变，谁都不能改！既然是修饰变量，那么变量的类型就可以丰富多彩，int 啊，char 啊，只要 C 认识的都可以；但是#define 就不可以了，在预处理阶段缺乏类型检测机制，有可能会出错。还有就是变量可以 extern，但是#define 就不可以。貌似 const 还可以节省 RAM，这个我倒是没有考证过。至于 const 的用法和作用，有很多，我会总结后发上来。

20、C 语言的 volatile 的含义是什么。使用时会对编译器有什么暗示。

终于最后一题了，容易么……如果这个测试是一个关于嵌入式的，那么这道题非常重要！！从词面上讲，volatile 的意思是易变的，也就是说，在程序运行过程中，有一些变量可能会被莫名其妙的改变，而优化器为了节约时间，有时候不会重读这个变量的真实值，而是去读在寄存器的备份，这样的话，这个变量的真实值反而被优化器给"优化"掉了，用时髦的词说就是被"和谐"了。如果使用了这个修饰词，就是通知编译器别犯懒，老老实实去重新读一遍！可能我说的太"通俗"了，那么我引用一下"大师"的标准解释：

volatile 的本意是"易变的" 。

由于访问寄存器的速度要快过 RAM,所以编译器一般都会作减少存取外部 RAM 的优化，但有可能会读脏数据。当要求使用 volatile 声明的变量的值的时候，系统总是重新从它所在的内存读取数据，即使它前面的指令刚刚从该处读取过数据。而且读取的数据立刻被保存。

精确地说就是，优化器在用到这个变量时必须每次都小心地重新读取这个变量的值，而不是使用保存在寄存器里的备份。

下面是 volatile 变量的几个例子：

1). 并行设备的硬件寄存器（如：状态寄存器）

2). 一个中断服务子程序中会访问到的非自动变量(Non-automatic variables)

3). 多线程应用中被几个任务共享的变量

嵌入式系统程序员经常同硬件、中断、RTOS 等等打交道，所用这些都要求 volatile 变量。不懂得 volatile 内容将会带来灾难。

好了，答完了，也不知道标准答案是什么。如果有达人看到皱眉头的话，千万停下来，浪费您宝贵的几分钟时间指点一二，不胜感激！

docin 豆丁
www.docin.com